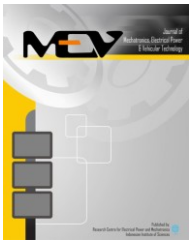


Mechatronics, Electrical Power, and Vehicular Technology 05 (2014) 27-36

Mechatronics, Electrical Power, and Vehicular Technology

e-ISSN: 2088-6985
p-ISSN: 2087-3379
Accreditation Number: 432/Akred-LIPI/P2MI-LIPI/04/2012

www.mevjournal.com

A TRAJECTORY GENERATION METHOD BASED ON EDGE DETECTION FOR AUTO-SEALANT CARTESIAN ROBOT

Eka Samsul Ma'arif [a,b,*], Endra Pitowarno [a], Rusminto Tjatur W [a]
[a] Electronic Engineering Polytechnic Institute of Surabaya
Jalan Raya ITS Sukolilo, Surabaya 60111
[b] Astra Manufacturing Polytechnic
Gaya Motor Raya No. 8, Astra International, Sunter, North Jakarta

Received 7 November 2013; received in revised form 12 February 2014; accepted 16 February 2014
Published online 23 July 2014

Abstract

This paper presents algorithm ingenerating trajectory for sealant process using captured image. Cartesian robot as auto-sealant in manufacturing process has increased productivity, reduces human error and saves time. But, different sealant path in many engine models means not only different trajectory but also different program. Therefore robot with detection ability to generate its own trajectory is needed. This paper describes best lighting technique in capturing image and applies edge detection in trajectory generation as the solution. The algorithm comprises image capturing, Canny edge detection, integral projection in localizing outer most edge, scanning coordinates, and generating vector direction codes. The experiment results show that the best technique is diffuse lighting at 10 Cd. The developed method gives connected point to point trajectory which forms sealant path with a point to next point distance is equal to 90° motor rotation. Directional movement for point to point trajectory is controlled by generated codes which are ready to be sent by serial communication to robot controller as instruction for motors which actuate axes X and Y directions.

Keywords: canny edge detection, integral projection, scanning the coordinate, vector direction code.

I. INTRODUCTION

Sealant process is one of production stages which has purpose to cover machine surface with adhesive liquid. Specific surface is the edge of machine block that liquid prevents air exchangein machine space. It has been developed using Cartesian robot that its movement programmed to follow the specific surface to be covered.

A Cartesian robot (also called linear robot) is an industrial robot whose three principal axes of control are linear (i.e. they move in a straight line rather than rotate) and are at right angles to each other[1]. High precision, low reject product and rapid process obtained by using it. But, large amount of machine models or new machine models cause new problem, a programmer has to do re-programming and re-calibration in change every models. So, we need fast adaptive and flexible robot for many models of machine.

Basically, every robot works with sequential processes that is defined by computer or controller trough codes, programming languages or pictures. Common Cartesian robots have a machine control unit (MCU) which inputs a numerical program to control the behavior and movements of all the parts of the machine. Currently numerical programs interpreted by MCUs are formed by an assembler-like code which is divided into single instructions called G-codes. Another way is using geometric models, they are generated in a Computer Aided Design (CAD)-system and then exported to Computer Aided Manufacturing (CAM) software [2]. Both codes and geometric models are offline conventional methods which require re-programming in trajectory change.

Online or sensor based trajectory generation considers the motion of a system to be dependent on the sensor at the robot or movement pattern, which means that the motion is directly modified based on the new command in every iteration. Pitowarno proposed an alternative scheme called Knowledge-Based Trajectory Error Pattern Method (KBTEPM) to suppress the trajectory track error of the AFC (Active Force Control) scheme. The knowledge is developed from the

* Corresponding Author. Phone: +62-85691655860
E-mail: ekasamsul@student.eepis-its.edu

doi: 10.14203/j.mev.2014.v5.27-36

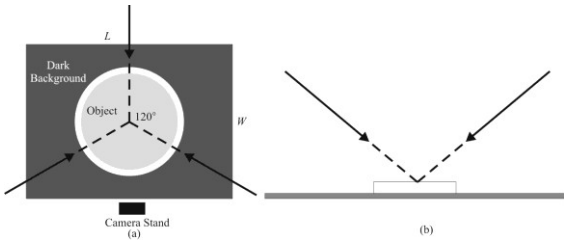


Figure 1. Direction lighting (a) Top view, (b) Side view

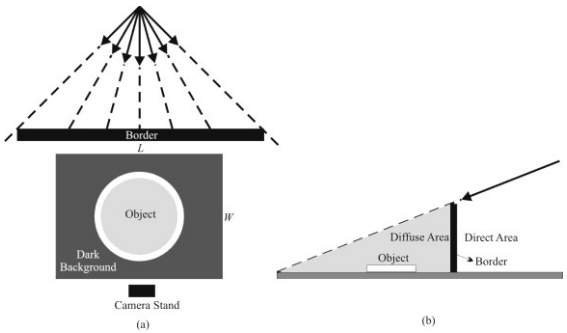


Figure 2. Diffuse lighting (a) Top view, (b) Side view

trajectory track error characteristic based on the previous experimental results of the crude approximation method. His method was success to control two-link robot arm [3].

A new approach is using captured image. Images treated with image processing can generate information to give instruction as the robots trajectory. Thus, computer vision becomes important in robotics and industries. Takarics and Szemes presented a new way to determine the trajectory for welding robots based on the intelligent space concept. The system uses two cameras and edge detection with other image processing algorithms to find the welding path in the image. It's three dimensional position is obtained by stereo vision and then it is transformed to the robot language [4].

Another online trajectory research was developed by Bojan Kuljic, he has developed autonomous mobile robot to find path in unknown indoor environment based on edge detection which combined with infra red distance as measuring sensor. 3D feature of environment was extracted using 2D image detection in determining object in front the robot. Lines detected was considered as obstacle then measured distance by infra red, so robot can generate avoidance algorithm to find other path for its movement [5].

Computer vision allows robot not only to determine environment but also to recognize specific object. An object recognizing system in remote control weapon station has been developed by Midriem Mirdanies. The research applies Scale Invariant Feature Transform (SIFT) and Speeded Up Robust Features (SURF) to define the number of vehicle's keypoint and descriptor, then stores it to database as registered object. The data from database was compared to capturing image from camera in real time object recognition. The name of captured object will be displayed if has suitable data with registered object [6].

This paper presents offline trajectory generation using captured image. Previous researches above had proved that computer vision has essential part in trajectory generation and environment recognizing. The objective of this paper is to find edge of the machine then processes it to trajectory, so edge detection is used. Edges in images are areas with strong intensity contrasts – a jump in intensity from one pixel to the next. Several algorithms exist, different method work better under different condition and goal.

This paper focuses on Canny detection method because of the good performance and detail detected result [7]. Beside, Canny edge detector is widely considered to be the standard edge detection algorithm in the industry [8]. Many publications have written about Canny edge detection method, such as Prof. Prem Kalra at his Lecturer Classroom who describes step by step of Canny edge detection method clearly [9]. Pixel coordinates from detected edge becomes main information for generating trajectory from one pixel point to another as axis movement of the Cartesian robot.

Section II explains about image preparation, capturing technique, and edge detection. Canny edge detection is used to detect edge of machine block then it is localized from the background. The output of Canny edge detection still contain various pixel of edge, so another algorithm needed go get only one specific edge. Integral Projection applied to detect outer edge boundary and get the outer most pixel. Finally, edge is scanned in order to get the coordinates. Section III shows experiment result and data acquisition in generating accurate trajectory codes. Section IV presents the research conclusion and further research.

II. METHODOLOGY

A. Image Preparation

Main step in image preparation is lighting technique. The quality and appropriateness of lighting are critical aspects for creating a robust vision inspection [10]. Lighting techniques can be categorized as direct and diffuse lighting. Both direct and diffuse are used in this research. Dark background is needed to avoid reflection that becomes noise in image processing. Lighting

which spread evenly is used to avoid shadow in machine block. Lighting directions are shown in Figure 1 and Figure 2.

Figure 1 is direct lighting where 3 light sources are used directly to machine. Figure 2 is diffuse lighting with object placed in diffuse area. Both direct and diffuse are keep light spread evenly around the background and object to avoid shadow. Light intensity at object is controlled and measured by a Luxmeter.

Secondary step is capturing image. Cartesian robot has main duty to do sealant process which its movement follows edge of machine accurately, therefore image must provide accurate information including absolute 2D projection and image scale. Image scale has a role in dimensional ratio between image and real object.

Background has been marked to keep every captured image always has same size (L and W). L and W are length and width of real dimension respectively. l and w are length and width of scaled dimension respectively. So, 1 pixel in image represents certain millimeter in real dimension called ratio (r).

$$r = \frac{L}{l} \tag{1}$$

Figure 3 shows how to capture machine block (object). Background area represents real dimension (in mm) and image is scaled dimension (in pixel). A pocket camera Canon Ixus 220 HS is used to capture machine image with dimension 4:3 in long and width. Captured image by the camera is stored as a red, green and blue (RGB) image. RGB image stores the value of each pixel in 3 channels but it could take too much time and consume too much memory for computer, therefore RGB image has to be converted to grayscale image. Almost 90% of edge information in a color image can be found in the corresponding grayscale image [11].

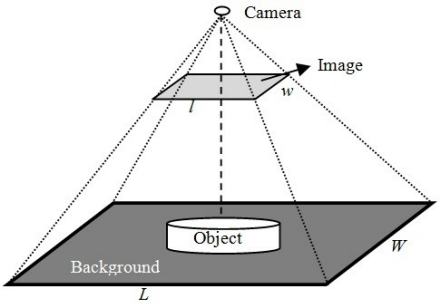


Figure 3. Capturing technique

B. Canny Edge Detection

The Canny edge detection algorithm is known to many as the optimal edge detector. John F. Canny was very successful in achieving his goal which his ideas and methods can be found in his paper, "A Computational Approach to Edge Detection". The aim was to develop an algorithm that is optimal to the following steps [9]:

1) Smoothing using Gaussian Filter
2) Finding Gradients using Sobel Operator
3) Non-maxima Suppression to sharpen the edge
4) Thresholding and Edge Tracking by Hysteresis to separate between strong and weak edges.

Edge detection program was developed in Microsoft Visual Studio 6.0 and Open CV Libraries. Figure 4 shows steps of Canny edge detection. Blurred image as the result of Gaussian filter kernel has purpose to reduce noise from input image. Then Sobel-operator and Non-Maxima Suppression are applied respectively in getting edge and sharpen it. Image still contains

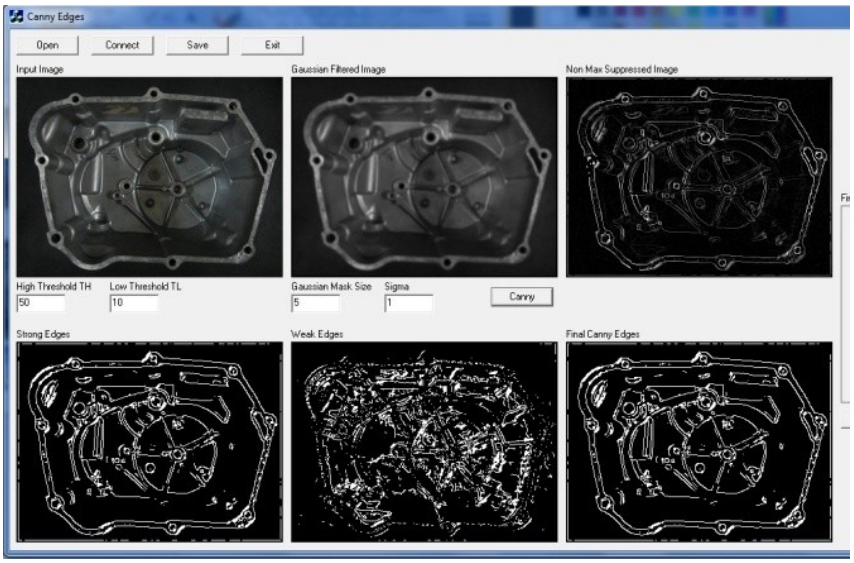


Figure 4. Canny edge detection steps

various intensity of edge at this step. Some may true edge but the rest is noise [12]. Only strong edge with certain value will be kept by using double threshold. Stronger edge than high threshold marked as strong edge, weaker edge than low threshold erased, then edge between high and low threshold assume as weak edge [13], [14]. So that strong edge can be isolated from weak edge then the final result of edge detection process is only image with strong edges.

C. Localize Main Edge

Final result from Canny edge detection still contain a lot of edges, whereas only outer most edge needed. To localize the outer most edge integral projection function is applied. Due to their simplicity and robustly, image integral projection functions have been used widely for the detection of the boundary between different image regions. Among them, the vertical and horizontal integral projection functions are most popular. Here, suppose $I(x,y)$ is the intensity of a pixel at location (x,y), the vertical integral projection function $IPFv(x)$ and horizontal integral projection function $IPFh(y)$ of $I(x,y)$ in intervals [y1, y2] and [x1,x2] can be defined respectively as:

$$IPF_v(x) = \int_{y1}^{y2} I(x,y)dy \quad (2)$$

$$IPF_h(y) = \int_{x1}^{x2} I(x,y)dx \quad (3)$$

The above two functions are used to detect the boundary of different image regions in the vertical and horizontal directions. Assuming PF is a projection function and ξ is a small constant. Thus, if the value of PF rapidly changes from z_0 to $(z_0+\xi)$, it indicates that z_0 lie on the boundary between two homogeneous regions. Using this approach, we can localize machine image from the background and detect the outer most side [15].

D. Finding Pixel Coordinate

The outer most edge obtained by applying Integral Projection. Then, every pixel coordinate must be found in order to generate point to point direction. Scanning process began by divides image frame into 4 quadrants [16], (Figure 5).

Scanning algorithm described below :

1) Scan quadrant *II* by increasing x coordinate form left to right pixel and find y with direction from bottom to top, note every pixel found.
2) Scan quadrant *I* by increasing x coordinate form left to right pixel and find y with direction from top to bottom, note every pixel found.
3) Scan quadrant *IV* by decreasing x coordinate form right to left pixel and find y with direction from top to bottom, note every pixel found.
4) Scan quadrant *III* by decreasing x coordinate form right to left pixel and find y with direction from bottom to top, note every pixel found.

Figure 6 is pixel coordinates illustration. Program will computer scanning algorithm start from most left pixel in *II* quadrant to continue right pixel from top coordinate to bottom. According to Figure 6, pixel coordinates list should be, 2,7 ; 3,6 ; 4,5 ; 4,4 ; 5,3 ; 6,3…… ; 3,12 ; 3,11 ; 3,10 ; 2,9 ; 2,8. First pixel is initial point to begin and last pixel is end point of sealant process. Pixel coordinate has difference with Cartesian coordinate in *Y* axis, so *Y* axis has to be multiplied by -1 to equate perception.

E. Generating Trajectory

True edge pixels will always be associated with other edge pixels. So, initial pixel to next pixel always has 1 pixel difference (+1 or -1) in

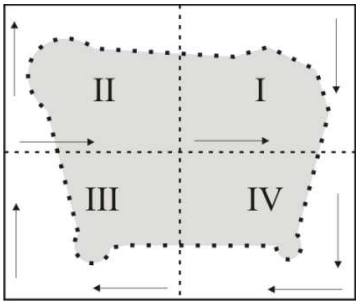


Figure 5. Scanning algorithm illustration

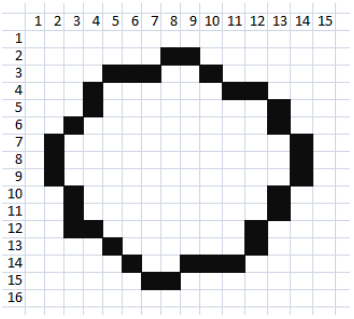


Figure 6. Pixel coordinates illustration

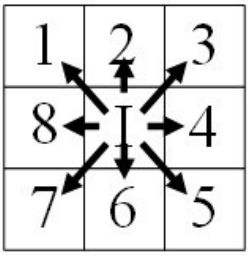


Figure 7. Vector direction code

Table 1.
Vector direction code logic

	ΔY= 1	ΔY= 0	ΔY= - 1
ΔX= - 1	1	4	7
ΔX= 0	2	0	6
ΔX= 1	3	8	5

Table 2.
Vector direction code to dc motor rotation

Code	Motor X	Motor Y
0	Off	Off
1	Reverse	Forward
2	Off	Forward
3	Forward	Forward
4	Forward	Off
5	Forward	Reverse
6	Off	Reverse
7	Reverse	Reverse
8	Reverse	Off

vertical, horizontal or both of them. Generating vector direction code uses Dijkstra algorithm (shortest path). Direction of displacement from initial pixel to next pixel determines the code, Figure 7.

Program computes algorithm from initial pixel (1) to next pixel (2) by subtraction operation y_1-y_0 and x_1-x_0 in Cartesian coordinate. Possible result of both y_1-y_0 and x_1-x_0 are 1,-1 or 0, and explained to appropriate direction code in Table 1. According to Figure 6, vector direction code generated is 3, 3, 2, 3, 4, 4 ……, 8, 2, 2, 1, 2, 2. Directional codes will be saved then sent to motor controller. Direction code is converted to motor *X* and *Y* rotation as shown in Table 2.

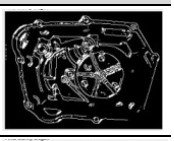
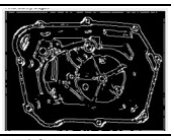
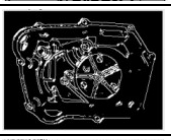
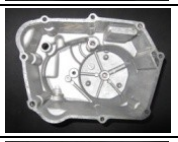
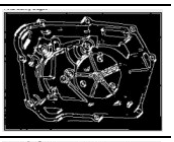
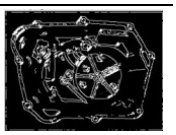
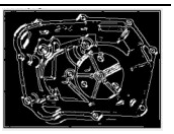
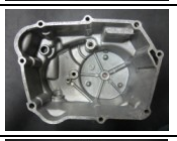
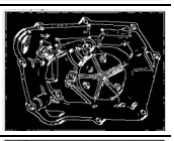
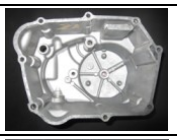
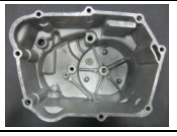
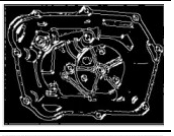
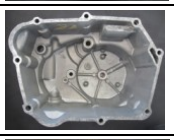
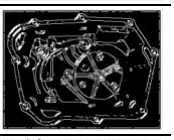
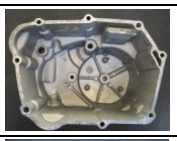
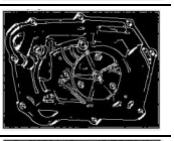
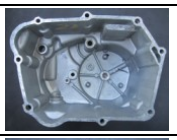
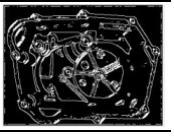
III. Result and Discussion

A. Lighting Technique and Edge Detection

Capturing process has been done indoor in order to simplify lighting adjustment and light intensity was measured by Luxmeter. L and W were always kept in 40 cm and 30 cm to get stable image scaling by adjusting distance from camera to the object (machine). Captured images are stored at 320x240 pixels dimension, so every pixel of the image represents r =1.25 mm in real object. Capturing set up was set as shown by Figure 8.

Direct lighting and diffuse lighting were applied in capturing process with various intensity (I) in Candela, start from 10 Cd to 3,000 Cd. Low intensities among 10 Cd to 400 Cd was obtained from 50W tab lamp configuration and

Table 3.
Captured image in various intensity and lighting

I (Cd)	Light	Image	Edge
10	Direct		
10	Diffuse		
30	Direct		
30	Diffuse		
70	Direct		
70	Diffuse		
200	Direct		
200	Diffuse		
300	Direct		
300	Diffuse		
400	Diffuse		
800	Diffuse		
1,000	Diffuse		
3,000	Diffuse		

higher one was produced by 500 W halogen lamp. Table 3 shows captured images and detection results in various intensities with two different lighting techniques.

B. Edge Boundary and Finding Pixel Coordinate

The captured image with intensity of 10 Cd and diffuse lighting is used to localize boundary in getting outer most edge shown in Figure 9. Figure above shows that outer most pixels are found successfully although contains some noise. Then, scanning algorithm (Figure 5) is applied to find pixel coordinates.

Pixel coordinate which found in scanning process is read in MATLAB as matrix, then multiplied with [1 -1] to get same orientation in Plotting Cartesian coordinate. Plotted coordinate has main purpose to verify between scanning accuracy with real capturing object, Figure 10.

Edge detected may contain noise called disperse pixel or jumping coordinate which y_1-y_0 and x_1-x_0 greater than 1 or less than -1 (marked in red), therefore filtering has to be applied. Changes values of x form a point to next point are always 1, 0 or -1 because of increasing and decreasing scanning algorithm at Figure 5, but not in y. Jumping coordinate has drastic changes in y, Figure 11.

Filter has been design to correct x and y when jumping pixel was happened. Filter will replace x_1 and y_1 with x_c and y_c (pixel connector to next pixel) as shown in Figure 12 [17].

Pixel Connector for (x_0,y_0) to (x_2,y_2) :

- If $x_2 - x_0 = 2$, or $x_2 - x_0 = 1$, then $x_c = (x_0 + 1)$
- If $x_2 - x_0 = -2$, or $y_2 - y_0 = -1$, then $x_c = (x_0 - 1)$
- If $y_2 - y_0 = 2$ or $y_2 - y_0 = 1$,

(a) (b)

Figure 8. Capturing; (a) Direct; (b) Diffuse

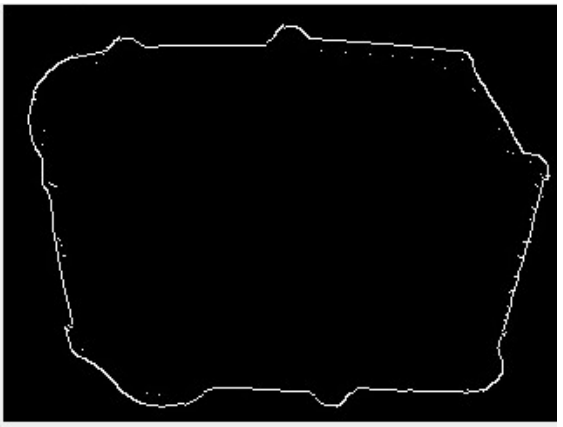


Figure 9. Localized edge boundary

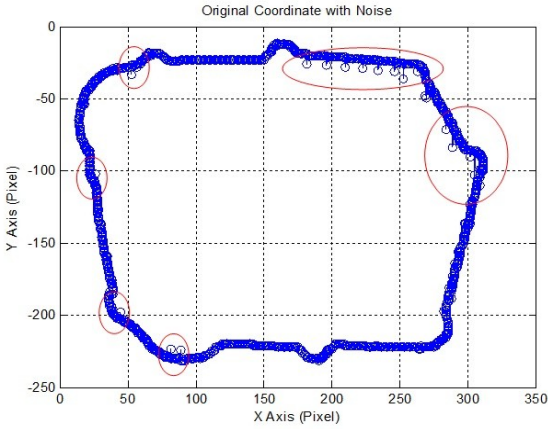


Figure 10. Original coordinate with noise

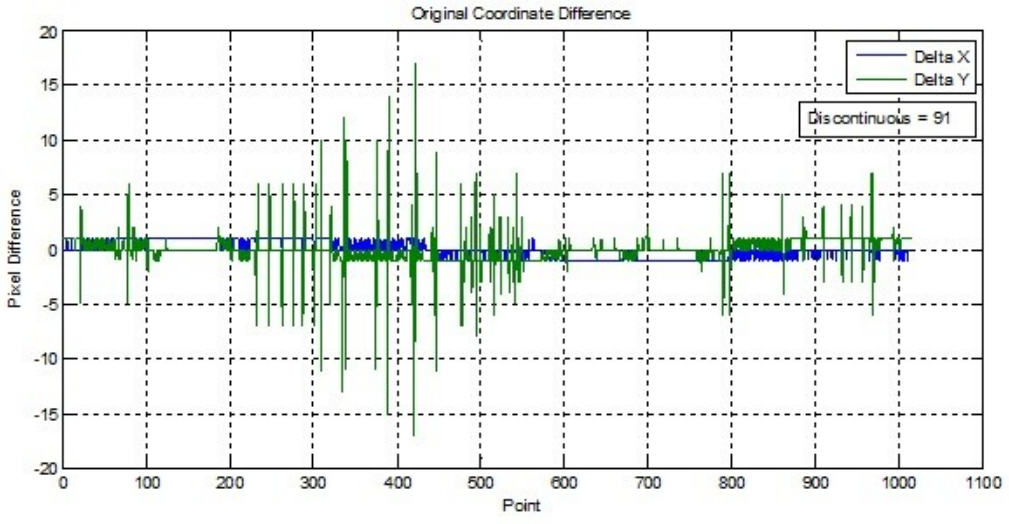


Figure 11. Original coordinate difference

Table 4.
Filtered coordinate

Point	Pixel Coordinate		Cartesian Coordinate	
	X	Y	X	Y
1	14	70	14	-70
2	14	69	14	-69
3	14	68	14	-68
4	14	67	14	-67
5	14	66	14	-66
6	14	65	14	-65
7	14	64	14	-64
8	14	63	14	-63
9	14	62	14	-62
10	14	61	14	-61
…….	…….	…….	…….	…….
…….	…….	…….	…….	…….
1,007	17	79	17	-79
1,008	16	78	16	-78
1,009	16	77	16	-77
1,010	16	76	16	-76
1,011	16	75	16	-75
1,012	15	74	15	-74
1,013	15	73	15	-73
1,014	15	72	15	-72
1,015	15	71	15	-71
1,016	15	70	15	-70

then $y_c = (y_0 + 1)$

- If $y_2 - y_0 = -2$ or $y_2 - y_0 = -1$, then $y_c = (y_0 - 1)$

Designed filter replaces jumping pixels so the differences are located from -1 to 1 as shown in Figure 13. Table 4 shows filtered in pixel coordinate and final result in Cartesian coordinate. Figure 14 shows plotted filtered coordinate in MATLAB graphic. Combined finding and filter algorithm successfully find continuous pixel coordinate which is main idea in point to point trajectory.

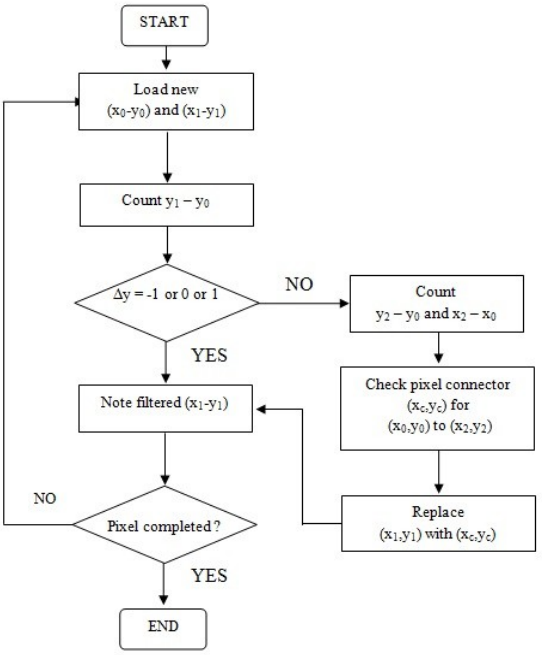


Figure 12. Flowchart filter pixel coordinate

C. Generating Trajectory from Pixel Coordinate

Pixel coordinate which plotted in figure 11 has sequence start from initial pixel to next pixel up to last pixel in clockwise direction. Initial pixel is (14,-70) as the most left pixel in quadrant II and the last is (15,-70). Deviation from pixel n to pixel n+1 is calculated to generate vector direction code. Generated code from Cartesian coordinate in Table 4 is saved in .txt file and the code list is shown in Tabel 5. Figure 15

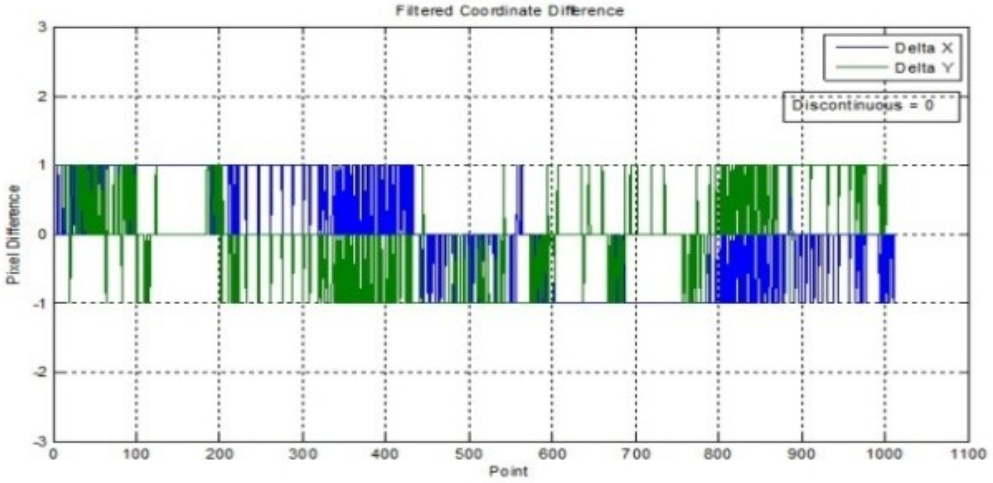


Figure 13. Filtered coordinate difference

represents point to point trajectory from generated vector direction codes. Generated codes are point to point trajectory which form sealant path, Figure 16.

Codes will be sent using serial communication from PC to controller robot in controlling axes movement which is actuated by DC motor. Cartesian robot has simple movement direction such as horizontal, vertical and diagonal with constant speed, and the position control functions

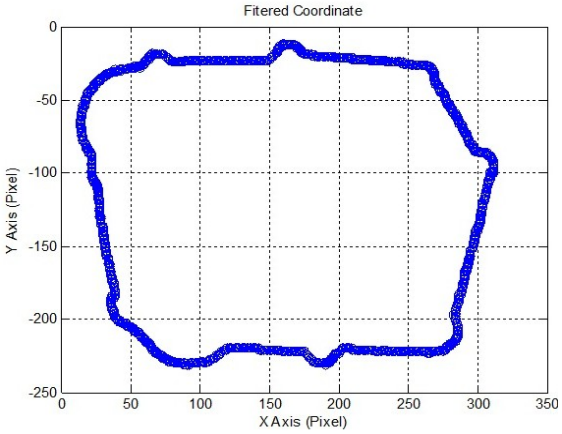


Figure 14. Filtered coordinate

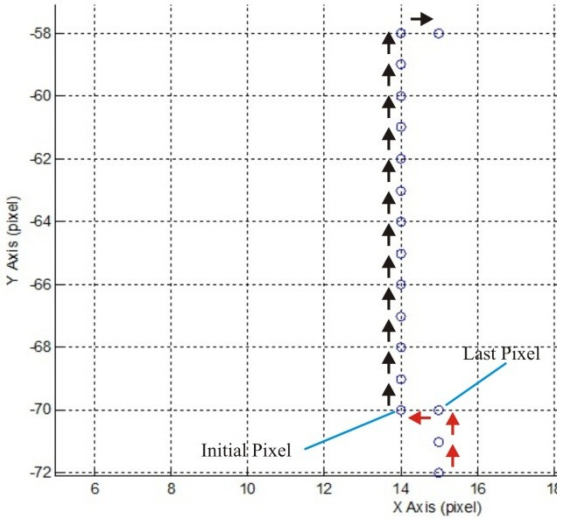


Figure 15. Point to point trajectory

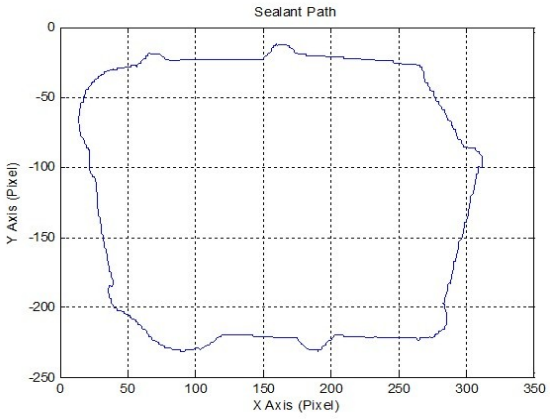


Figure 16. Point to point trajectory

Table 5.
Direction code list

Point	Direction Code to Next Point
1	2
2	2
3	2
4	2
5	2
6	2
7	2
8	2
9	2
10	2
…….	…….
…….	…….
1,007	2
1,008	1
1,009	2
1,010	2
1,011	2
1,012	1
1,013	2
1,014	2
1,015	2
1,016	2

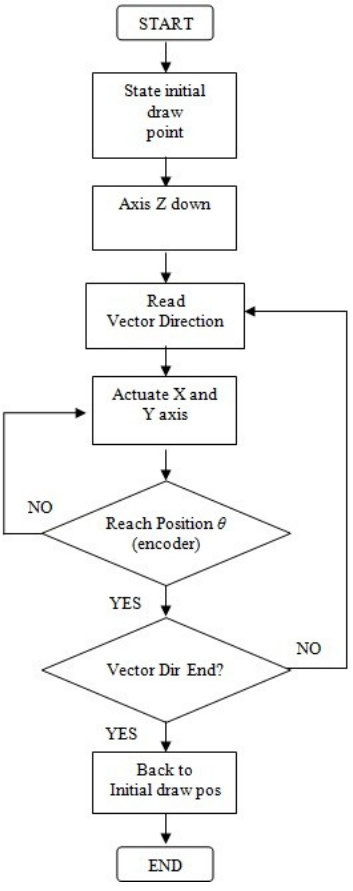


Figure 17. Flow chart axis movement

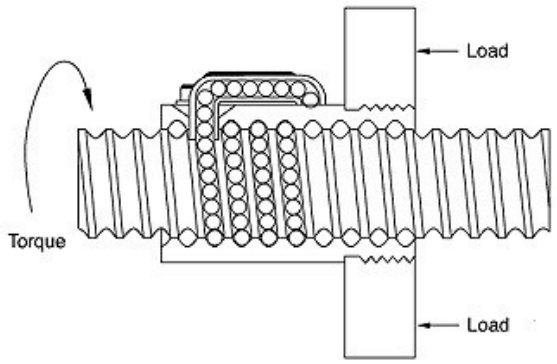


Figure 18. Stroke and ball screw of the axis

Figure 19. Cartesian robot

only to follow the direction code in point to point manner. Figure 17 shows flow chart robot axis movement in following trajectory generated by vector direction code. The rotation angle of DC Motor rotation θ is calculated according to correlation between pixel displacement and stroke of the axis.

Figure 18 shows stroke and ball screw. This robot uses stroke with 5 mm lead, it means 1 full motor rotation bring 5 mm displacement in *X* or *Y*. Thus, 1.25 mm displacement from 1 pixel to another neighbor pixel needs 0.25 motor rotation or 90°. Motor will be controlled at certain constant speed and the step reference value along each axis is 90° in every movements. The Cartesian robot is shown at figure 19.

IV. CONCLUSIONS

Lighting has main role in capturing process to get good image, so that edge can be detected accurately. By the experiments, diffuse lighting technique with evenly spreading light gives better results than direct one both high intensity and low intensity. But, high light intensity frequently causes reflection and shadow that disguise the real edge at detection process.

Designed edge detection algorithm has good performance in intensity 10 Cd – 3,000 Cd with diffuse lighting technique, but the best result is 10 Cd. The outer most edge pixel has been obtained by applying integral projection and its coordinates have been successfully founded by scanning algorithm. 1,016 continuous coordinates are successfully found and sealant path has accurate shape with captured machine which was verified with MATLAB graphic plot. It consists of connected point to point trajectory which the distance a point to next point equal to 90° motor rotation in X axis, Y axis or both of them. Directional movement for point to point trajectory is controlled by generated codes.

In further research, codes are sent to robot controller using serial communication as instruction for axis movement to do sealant process. Each axis is actuated by DC motor that its rotation position controlled according to certain direction and displacement.

ACKNOWLEDGEMENT

The author would like to thank the Electronic Engineering Polytechnic Institute of Surabaya for giving laboratories facilities for the research and Astra Manufacturing Polytechnic for providing financial support.

REFERENCES

[1] Y. Maddahi, "Reliability and Quality Improvement of Robotic Manipulation Systems," *WSEAS Transactions on Systems and Control*, vol. Volume 6, pp. 339-348, 2011.

[2] S. Brell-Cokcan, "A New Parametric Design Tool for Robot Milling," *ACADIA*, 2010.

[3] E. Pitowarno, *et al.,*"Knowledge-Based Trajectory Error Pattern Method Applied To An Active Force Control Scheme," *IIUM Engineering Journal*, vol. 3 No.1, 2002.

[4] B. Takarics and P. T. Szemes, "Super flexible Welding Robot Based on the Intelligent Space Concept," presented at the 7th International Symposium of Hungarian Researchers on Computational Intelligence, 2008.

[5] B. Kuljic, "Pathfinding Based on Edge Detection and Infrared Distance Measuring Sensor," *Acta Polytechnica Hungarica*, vol. 6 No.1, pp. 103-116, 2009.

[6] M. Mirdanies, *et al.,* "Object Recognition System In Remote Controlled Weapon Station Using Sift And Surf Methods," *Journal of Mechatronics, Electrical Power and Vehicular Technology*, vol. 04, No. 2, pp. 99-108, 2013. DOI : 10.14203/j.mev.2013. v4.99-108

[7] E. S. Maarif, "Applied Canny Edge Detection in Trajectory Planning for Auto-

Sealant Cartesian Robot," *in Indonesian Symposium on Robot Soccer Competition 2013*, Semarang, Indonesia, 2013.

[8] E. Nadernejad, "Edge Detection Techniques: Evaluations and Comparisons," *Applied Mathematical Sciences*, vol. 2, No. 31, pp. 1507 – 1520, 2008.

[9] P. Kalra. "*Canny Edge Detection,*". Lecturer Classroom. Department of Computer Science and Engineering. India Institute of Technology, New Delhi. 2009. [Online]. Available: www.cse.iitd.ernet.in/~pkalra/csl783/canny.pdf

[10] Y. Surya, "Machine Vision Implementation in Rapid PCB Prototyping," *Journal of Mechatronics, Electrical Power, and Vehicular Technology*, vol. 02, No. 2, pp. 79-84, 2011. DOI : 10.14203 /j.mev.2011. v2.79-84

[11] S. Dutta and B. B. Chaudhuri, "A Color Edge Detection Algorithm in RGB Color Space," in *International Conference on Advances in Recent Technologies in Communication and Computing*., 2009, pp. 337-340.

[12] M. Juneja and P. S. Sandhu, "Performance Evaluation of Edge Detection Techniques for Images in Spatial Domain," *International Journal of Computer Theory and Engineering*, vol. 01, No. 5, pp. 614-621, December 2009.

[13] G. T. Shrivakshan and C. Chandrasekar, "A Comparison of various Edge Detection Techniques used in Image Processing," *International Journal of Computer Science Issues (IJCSI)*, vol. Vol. 9, Issue 5, No. 1, pp. 269 -276, September 2012.

[14] P. Zhou, "An Improved Canny Algorithm for Edge Detection," *Journal of Computation Information Systems*, pp. 1516-1523, 2011.

[15] G. J. Mohammed, "Eyeball Localization Based on Angular Integral Projection Function," *Slovenian Society Informatica*, vol. 33 Issue 4, pp. 475-480, November 2009.

[16] S. P. Khandait, *et al.,* "Comparative Analysis of ANFIS and NN Approach for Expression Recognition using Geometry Method," *International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 2, Issue 3, March 2012.

[17] L. Xiao-Yan and C. Yan-Li, "Application of Dijkstra Algorithm in Logistics Distribution Lines," in *Third International Symposium on Computer Science and Computational Technology (ISCSCT '10)*, Jiaozuo, P. R. China, August, 2010, pp. 048-050.